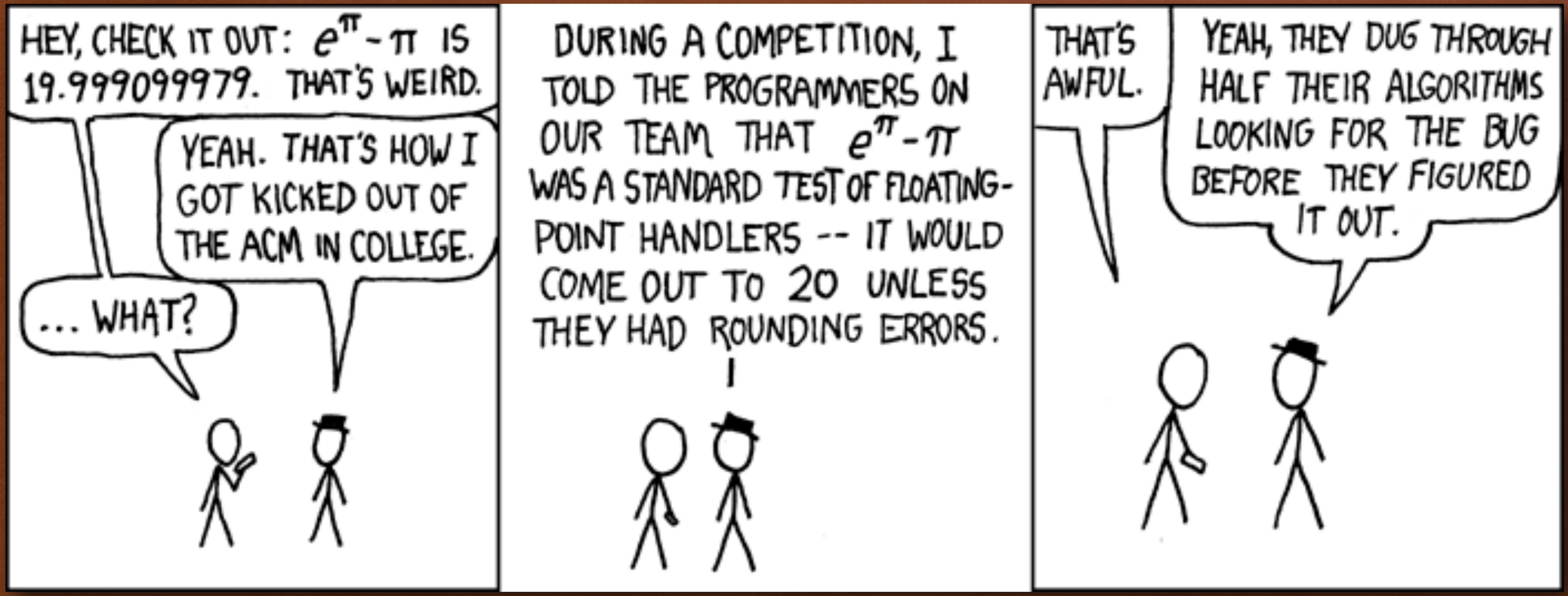
THE BUILDING BLOCKS:
MORE BINARY ARITHMETIC, NEGATIVE VALUES AND FLOATING POINT



LEARNING OBJECTIVES

- Represent signed binary numbers in different formats
- Use two's complement as the standard integer representation
- ~~Do multiplication and division of binary values~~
- ~~Use shifts to multiply and divide~~
- ~~Binary fractions~~
- The above objectives are now in lecture 4

SIGNED INTEGERS + AND -

- With n bits, we can distinguish 2^n unique values
  - assign about half to positive integers (1 to 2^{n-1}-1)
    and about half to negative (-2^{n-1}+1 to -1)
  - that leaves two values: one for 0, and one extra
- Positive integers
  - just like unsigned, but zero in the **Most Significant Bit** (MSB)
    `00101 = 5`
- Negative integers
  - Sign-Magnitude (or Signed-Magnitude) – set MSB bit to show negative, other bits are the same as unsigned
    `10101 = -5`
  - One's complement – flip every bit to represent negative

```
00101 = +5
11010 = -5
```

- In either case, MS bit indicates sign: 0=positive, 1=negative
- https://www.youtube.com/watch?v=lKTsv6iVxV4

SIGNED INTEGERS

- Using either method we have to know which bit is the MSB.
- This means that for signed values we must know how many bits there are in the number. So all questions will specify the number of bits we are using to represent the number.
    - A question like convert -347 into binary cannot be answered uniquely unless we specify the format.
    - Normally we deal with full bytes so our integers are commonly 8, 16, 32 or 64 bits long, but we could ask questions where the number is not one of these values.
    - With a finite number of bits for our answers we are commonly going to have a **carry** out from the MSB when we do arithmetic.

WHAT VALUE DO WE HAVE?

- If we have the number `11001` what value does it represent?
- If it is a 5 bit signed magnitude number we see that it is negative because the 5th bit is `1`.
    - the magnitude is `1001 = 9`, the original was `-9`.
- If it is a 5 bit one's complement number the number is still negative but to work out the value we have to flip all of the bits.

`11001`
`00110 = 6` so the number was `-6`.

PROBLEMS

- Say we have 4-bit signed values what do the following numbers represent in sign-magnitude?

  `0000` and `1000`

- and these in one's complement?

  `0000` and `1111`

- Also do the following addition in sign-magnitude (4-bit)

  `0010 + 1011` what should the answer be?

- and this one in one's complement (4-bit)

  `0100 + 1100` what should the answer be?

REVISION

- Show decimal -12 as 5 bit signed magnitude
- Show decimal 20 as 6 bit one’s complement
- Show decimal -20 as 6 bit one’s complement

TWO'S COMPLEMENT

- With a slight change to the one's complement representation of negative integers we can solve both of these problems.
  - This also makes it trivial to do subtraction on computers using the same circuits we use for addition (see following lectures).
- If number is positive or zero,
  - normal binary representation, zeros in upper bit(s)
- If number is negative,
  - start with positive number
  - flip every bit (i.e., take the one's complement)
  - then add one

TWO'S COMPLEMENT EXAMPLES

- For 5 bit numbers

```
00101 (5)          01001 (9)
11010  flip        10110  flip
    1                  1
-----              -----
11011  (-5)        10111 (-9)
```

- And if we add 5 + -5 or 9 + -9 together what do we get?

```
  00101              01001
  11011              10111
  -----              -----
1 00000            1 00000
```

- The carry is ignored and the 5-bit answer is zero

HOW DOES THAT WORK?

- Why can we ignore the carry out when doing the additions on the previous slide?
- Because two's-complement is using modulo (or clock arithmetic), when we wrap around past all the 1s the answer is still correct.
- e.g with 4-bit numbers

```
1010 + 0110 = 1 0000
```

Can be interpreted as unsigned 10 + 6 = 16 (we need the carry here) or as 2's complement -6 + 6 = 0 (we don't use the carry here)

TWO'S COMPLEMENT RANGE

- Range of an n-bit number: -2^{n-1} through $2^{n-1} – 1$.
  - The most negative number (-2^{n-1}) has no positive counterpart.
    - e.g. n = 8, the range is -2^7 (-128) to 2^7 - 1 (+127)

-2^3	2^2	2^1	2^0	
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7

-2^3	2^2	2^1	2^0	
1	0	0	0	-8
1	0	0	1	-7
1	0	1	0	-6
1	0	1	1	-5
1	1	0	0	-4
1	1	0	1	-3
1	1	1	0	-2
1	1	1	1	-1

- And you can see the wrap around from -1 back to 0. See pg 160.